УДК 519.713.1

С.М. Вороной, С.С. Еськов
Донецкий национальный технический университет, Украина
Украина, 83050, г. Донецк, пр. Богдана Хмельницкого, 84

Применение аппарата сетей Петри для проектирования обобщенной многопользовательской системы

S.M. Voronoy, S.S. Yeskov
Donetsk National Technical University, Ukraine
Ukraine, 83050, c. Donetsk, Bogdana Khmelnitskogo av.

Petri Net Approach for Design Generalized Multi-User System

С.М. Вороной, С.С. Єськов
Донецький національний технічний університет, Україна
Україна, 83050, м. Донецьк, пр. Богдана Хмельницького, 84

Застосування апарату мереж Петрі для проектування узагальненої багатокористувацької системи

В статье рассматривается задача построения многопользовательской системы с применением аппарата сетей Петри. Построен шаблонный проект многопользовательской системы при помощи высокоуровневого аппарата сетей Петри, описанного международным стандартом.
Ключевые слова: веб-приложение, сети Петри.

Petri net solution for design generalized multi-user system is described by the article. Pattern project of multi-user system was built using high-level Petri nets described by international standard.
Key words: web-application, Petri nets.

У статті розглядається задача побудови багатокористувацької системи шляхом застосування апарату мереж Петрі. Побудовано узагальнений проект багатокористувацької системи за допомогою мереж Петрі високого рівня, стандартизованого міжнародним стандартом.
Ключові слова: веб-додаток, мережі Петрі.

Целью работы является исследование возможностей применения математического аппарата сетей Петри для проектирования веб-приложений. Перед работой поставлены следующие **задачи**:

1. Обосновать возможность применения аппарата сетей Петри для проектирования многопользовательских систем.
2. Определить основные операции многопользовательской системы. Определить ограничения, которые накладываются на операции.
3. Построить обобщенную модель многопользовательской системы при помощи аппарата сетей Петри.

Под многопользовательской системой будем понимать информационную систему, которая характеризуется наличием N субъектов (пользователей) и M ресурсов. Причем, информационные связи возможны как между субъектом и ресурсом, так и непосредственно между субъектами. Описанная система, с одной стороны, может рассматриваться как многоагентная система с централизованным хранилищем данных и непредсказуемым поведением агентов. С другой стороны, данную систему можно определить как систему распределенного доступа с наличием информационных связей между пользователями.

Во избежание недоразумений, в данной работе будем придерживаться определения «многопользовательская система».

В последнее десятилетие произошло значительное развитие и популяризация открытых стандартов передачи данных и интерпретации команд программного обеспечения, таких, как протоколы TCP/IP, языки разметки SGML (в частности HTML [1]), языки программирования класса ECMAScript [2] (в частности JavaScript, ActionScript). Повсеместное внедрение этих стандартов повышает скорость доступа пользователей к информации, но также повышается сложность систем, обеспечивающих взаимодействие пользователей, так называемых веб-приложений [3]. Сегодня веб-приложения, по своей сложности, не уступают классическим приложениям для операционных систем. Становится вопрос о необходимости применения инструментов для проектирования таких систем [4], [5]. Популярные инструменты проектирования программного обеспечения, такие, как UML-диаграммы, диаграммы класса IDEF и другие, не обеспечивают выполнение требований, которые выдвигаются многопользовательскими системами [5]. В связи с этим актуальной научной задачей является применение других инструментов проектирования для эффективной разработки многопользовательских систем, а также создание новых методик проектирования программного обеспечения, обеспечивающих простое и гибкое моделирование асинхронных, распределенных, параллельных систем, которыми являются многопользовательские системы.

Математический и графический аппараты. Аппарат сетей Петри создан Карлом Петри для моделирования динамических, распределенных, параллельных систем [6]. Многопользовательские системы обладают свойствами, которые присущи системам, успешно моделируемым при помощи сетей Петри, поэтому применение данного математического аппарата при проектировании и моделирования многопользовательских систем является целесообразным. В работе сделана попытка применения аппарата сетей Петри для моделирования многопользовательских систем. Примененный в работе аппарат стандартизирован международным сообществом [7].

Сети Петри высокого уровня (HLPN) имеют следующую семантическую формализацию:

$$HLPN = (P, T, D; Type, Pre, Post, M_0),$$

где

– P – это ограниченное множество элементов, называемых *Состояниями*;

– T – это ограниченное множество элементов, называемых *Переходами*, T не пересекается с P ($P \cap T = \{\}$);

– D – это не пустое ограниченное множество не пустых множеств, где каждый элемент множества D обычно называют *Типом* (D задает множество типов меток, которые будут использованы в модели, например, $Int = \{\ldots, 0, 1, 2, \ldots\}$ – множество целых чисел, $A = \{a1, a2, a3\}$ – произвольный тип, состоящий из трех элементов);

– $Type : P \cup T \rightarrow D$ – это функция, связывающая Типы из множества D с Состояниями из множества P, и определяет режимы переходов;

– $Pre, Post: TRANS \rightarrow \mu PLACE$ – это наборы входящих и исходящих связей, где

$$TRANS = \{(t, m) \mid t \in T, m \in Type(t)\}$$
$$PLACE = \{(p, g) \mid p \in P, g \in Type(p)\}$$

– $M_0 \in \mu PLACE$ – это мультимножество, называемое начальной маркировкой сети, где $\mu PLACE$ – это множество мультимножеств, образованных над множеством P.

Формализация графического аппарата описания сетей Петри высокого уровня приведена далее.

$$HLPNG = (NG, Sig, V, H, Type, AN, M_0)$$

где $NG = (P, T; F)$ – это граф сети,

в котором

– P – это ограниченное множество элементов, называемых Состояниями;

– T – это ограниченное множество элементов, называемых Переходами, T не пересекается с P ($P \cap T = \{\}$);

– $F \subseteq (P \times T) \cup (T \times P)$ – это ограниченное множество направленных связей между *Состояниями* и *Переходами*, называемых *Дугами*;

– $Sig = \{S, O\}$ – это ограниченное множество элементов для описания операторов, используемых в сети,

где

– S – это ограниченное множество названий операторов, используемых при моделировании сети, например =, >, $\subseteq$ и т.п.;

– O – это ограниченное множество, элементом которого является арность оператора из множества S;

– V – это множество независимых переменных модели, непересекающееся с O;

– $H = \{Sh, Oh\}$ – это алгебра интерпретаций множества Sig, детальное описание приведено в [7];

– $Type : P \rightarrow Sh$ – это функция, связывающая Типы с Состояниями;

– $AN = (A, TC)$ – это пара аннотаций,

где

– $A : F \rightarrow TERM(O \cup V)$ – это функция, связывающая каждую *Дугу* с некоторой функцией, которая выполняется при переходе метки (одного из элементов элемента множества D) по этой дуге;

– $TC : T \rightarrow TERM(O \cup V)_{bool}$ – это функция срабатывания перехода, возвращающая булево значение;

– $M_0 \in \mu PLACE$, то же что M_0 в семантическом описании сети – это мультимножество, называемое начальной маркировкой сети, где это множество мультимножеств, образованных над множеством P.

Проектирование сети. Выделим операции, которые присущи всем многопользовательским системам, с целью последующего представления этих операций при помощи сети Петри.

1. Установление канала связи с ресурсом в хранилище данных (открытие документа, авторизация).
2. Уничтожение канала связи с ресурсом в хранилище данных (закрытие документа, прерывание сессии авторизации).
3. Создание нового ресурса.
4. Удаление ресурса.
5. Взаимодействие с пользователями (оповещение о проделанной текущим пользователем операции, получение оповещений о проделанных другими пользователями операциях).

Далее сделаем постановку требований к модели обобщенной многопользовательской системы. Обобщенная модель должна обеспечивать:

– выполнение всех перечисленных операций;

– обработку взаимных блокировок операций (например, нельзя делать удаление открытого ресурса и т.п.);

– расширяемость, с целью последующего применения модели как шаблона для проектирования реальных систем;

– наглядность;

На рис. 1 изображена сеть Петри, моделирующая описанные операции многопользовательской системы. Далее приведены некоторые тезисы, описывающие работу модели.

– позиция P0 расположена как крайняя слева и продублирована как крайняя справа на рис. 1, это сделано для удобства чтения модели;

– пользователи многопользовательской системы представлены как метки множества Users и изначально находятся в позиции P0, в процессе моделирования количество пользователей остается неизменным;

– ресурсы многопользовательской системы представлены как метки множества Resources и изначально находятся в позиции P1, в процессе моделирования количество ресурсов может изменяться;

– при успешном открытии ресурса (переход T3 – P2), метка из множества UR попадает в позицию P2;

– ошибка открытия отрабатывается переходом T1;

– в позиции P2 находятся метки из множества UR и могут быть интерпретированы как канал связи пользователя с ресурсом;

– если метка из множества UR уже присутствует в позиции P2, её повторное попадание в эту позицию блокируется переходом T3 – P2.

– закрытие ресурса осуществляется при срабатывании перехода T4, метка из множества UR покидает позицию P2;

– добавление нового ресурса осуществляется при срабатывании перехода T6 и возможно в любой момент времени;

– удаление ресурса осуществляется при срабатывании перехода T5 и возможно в том случае, если в множестве P2R не присутствует удаляемый ресурс.

– после редактирования ресурса (переход T5), происходит оповещение пользователей из множества P2U о редактировании, за исключением пользователя, сделавшего редактирование;

– операторы go, from использованы с целью не загромождать рисунок, таким образом соединены P1 – T9 и T8 – P1;

– переход P1 – T9 срабатывает при удалении ресурса, при помощи этого перехода удаляется метка ресурса из позиции P1, чтобы последующие операции были невозможны с удаленным ресурсом;

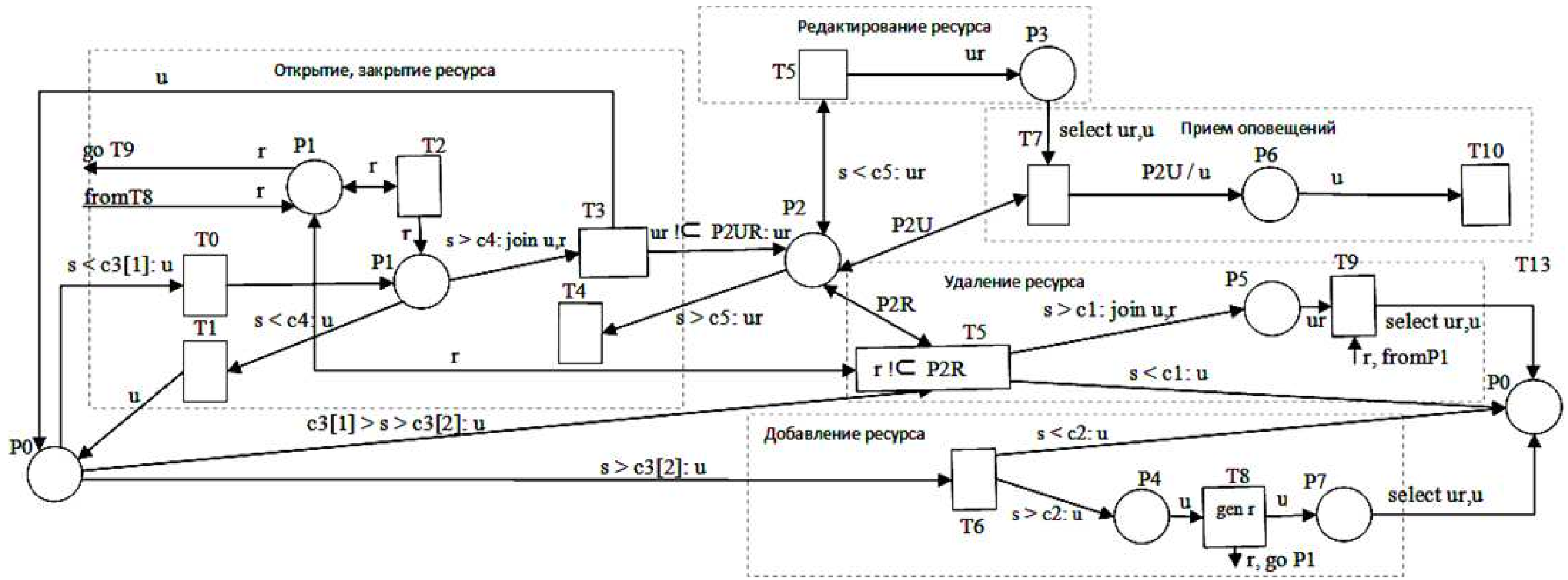
– переход T8 – P1срабатывает при добавлении нового ресурса, при помощи этого перехода добавляется метка ресурса в позицию P1, чтобы можно было произвести следующие операции с добавленным ресурсом;

– разрешение конфликтных ситуаций происходит при помощи генерации псевдослучайного числа s, которое сравнивается с константами (или элементами константных массивов) c1, c2, c3, c4, c5.

– операция gen создает новую метку, это необходимо при добавлении нового ресурса в систему;

– операция select создает множество из элементов, полученных путем выборки элементов заданного типа у элементов заданного множества (например, если K= {{a1,b1},{a2,b2}}, то select K,b будет равен {b1, b2});

– операция join создает множество из элементов, полученных путем объединения заданных множеств или элементов (аналог известной операции $\cup$, но с возможностью работы, как с множествами, так и с неделимыми элементами, например, join a,b будет равен {a, b});



Users = {user1, user2, user3, …}
Resources = {res1, res2, res3, …}
UR = {{user[i], res[j]}, {user[k], res[m]}, {user[t], res[r]}, …}
P2UR – элементы множества UR, которые находятся в позиции P2
P2U – подмножество элементов множества Users, образованное путем выделения составляющей user[i] из P2UR (select P2UR,u)
P2R – подмножество элементов множества Users, образованное путем выделения составляющей res[i] из P2UR (select P2UR,r)
ur – элемент множества UR
u – элемент множества Users
r – элемент множества Resources
s – случайное число 0…1, полученное по равномерному закону распределения
c1, c2, c3, c4, c5 – константа или массив констант 0…1, если массив, следующий элемент массива больше предыдущего

>,< – операции алгебраического сравнения
⊂ – операция проверки вхождения подмножества в множество
!⊂ – операция проверки отсутствия подмножества в множестве
/ - операция вычитания множеств
select m,e – выборка элементов e из множества m
join m1, m2 – создание нового множества путем соединения двух множеств или элементов
gen e – создание новой метки
условие : передаваемое значение – условный оператор
M0(P0) = 1'user1 + 1'user2 + 1'user3 + …
M0(P1) = M0(P2) = M0(P3) = M0(P4) = M0(P5) = M0(P6) = M0(P7) = M0(P8) = пустое множество

Рисунок 1 – Модель обобщенной многопользовательской системы, спроектированная при помощи сети Петри

Таким образом, в работе было промоделировано функционирование многопользовательской, асинхронной системы при помощи математического аппарата сетей Петри. Созданная шаблонная модель может быть расширена применительно к реальным системам путем добавления новых состояний, переходов, операторов, а также типов меток.

Литература

1. Спецификация языка HTML [Электронный ресурс]. – Режим доступа : http://www.w3.org/TR/html401/.
2. Спецификация языка ECMAScript HTML [Электронный ресурс]. – Режим доступа : http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf .
3. Беллиньясо М. Разработка Web-приложений в среде ASP.NET 2.0. / М. Беллиньясо. – 2007.
4. Hossny E. An Intelligent Web Service Workflow: A Petri Net Based Approach / E. Hossny, S. AbdElrahman and A. Badr // The Open Information Systems Journal. – 2010.
5. Moldt D. A Proposal for Petri Net Based Web Service Application Modeling / D. Moldt, S. Offermann, J. Ortmann // University of Hamburg. – 2004.
6. Murata T. Properties Analysis and Applications / T. Murata, Petri Nets. – 1989.
7. High-level Petri Nets – Concepts, Definitions and Graphical Notation, Final Draft // International Standard ISO/IEC 15909, Version 4.7.1, October 28, 2000.

RESUME

S.M. Voronoy, S.S. Yeskov

Petri Net Approach for Design Generalized Multi-User System

Functionality of multi-user asynchronous system was modeled in general case. Pattern design system was presented using Petri nets mathematical and graphical tools. Problems were resolved:

1) Repeat of resources re-opening problem.

2) The problem of resolving conflicts in case of asynchronous removing resources in multi-users system.

3) The issue of alerting users about changes in resource.

4) The problem of maintaining a parametric state of the network (number of tokens and their position in the structure of the network) up to date.

Studies have shown the applicability of the apparatus of Petri nets for modeling of multi-user systems. Created template model can be extended with respect to real systems by adding new states, transitions, operators and types of tokens.

Статья поступила в редакцию 25.04.2013.